Extending a Clinical Repository to Include Multiple Sites

Keith A. Marrs, MS and Michael G. Kahn, MD, PhD
Section of Medical Informatics, Washington University, St. Louis, Missouri

With the consolidation of health care organizations and services, a clinical repository comprising data from a single site is no longer sufficient. Individual patient data are now spread across multiple sites comprising a single enterprise. Users require an integrated view, or at least a common view, of these clinical data across multiple sites. Many issues arise when one tries to merge data from multiple, distinct organizations into an existing schema. We have addressed these issues while extending our clinical repository for Barnes Hospital with data from Jewish Hospital, both of which are members of the recently formed BJC Health System. We describe the architecture of our existing repository, approaches and issues in extending this repository to include multiple sites, and the specific issues we addressed in our system.

BACKGROUND

To control costs and increase efficiency, there has been a substantial increase in the mergers of hospitals, clinics, and other health care organizations. Specialized services often are consolidated at specific sites in these large organizations, resulting in the proliferation of a patient's data not only within a hospital but also across many hospitals and other facilities in the organization. A clinical repository with information from a single facility is no longer sufficient. A common view of data from multiple sites must be available for users to capture a patient's entire history. Also, decision-support applications that use a clinical repository easily can be ported to other facilities if data from these facilities can be mapped to the repository schema. However, moving from a repository that contains data for a single facility to one that contains data from multiple facilities is not always straightforward. Many syntactic and semantic issues must be addressed when mapping to the repository schema.

We have addressed these mapping issues while adding data from Jewish Hospital to our existing clinical repository. Our initial repository was developed to support both production and research decision-support applications that need an integrated view of clinical data from multiple heterogeneous distributed databases within Barnes Hospital.[1] Since the development of this clinical repository, BJC Health System was formed. BJC Health System includes 15 hospitals, including Barnes and Jewish Hospitals, and numerous other health care facilities. The formation of BJC Health System along with the popularity of our decision-support applications for infection control surveillance[2] and medication dose monitoring[3] have driven the requirement to include data from other hospitals in BJC Health System in our repository. Jewish Hospital was chosen because of its close association with Washington University School of Medicine and because Barnes and Jewish Hospitals share support staff for these decision-support applications.

In the remainder of the paper, we briefly describe the architecture of our repository, discuss alternative approaches for handling data from multiple sites, and provide specific details on our approach and issues we addressed in mapping data from Jewish Hospital to our repository schema. We conclude with a discussion of our approach and future plans for our repository.

REPOSITORY ARCHITECTURE

Our clinical data repository is implemented as a set of relations and constraints in a relational DBMS. Data are retrieved daily from Barnes Hospital information systems, mapped to a global schema, and merged with existing data in the repository.[1]

We chose to implement a physical instead of a logical repository for several reasons: technical and administrative access issues; predictability of performance; availability of the data; and ability to provide data abstractions. With a logical repository, data are not redundantly stored in another database. Instead, operations on the logical repository are mapped to operations on possibly multiple underlying systems. Results from these underlying systems are then transformed into the global schema of the logical repository and presented to the user. With a physical repository, we avoid immediate access issues such as distributed concurrency control, transaction management, and security of underlying systems.

A physical repository provides predictable performance since all operations are local and need not be transformed to operations on possibly multiple underlying systems. Also, the availability of data is dependent only on the availability of the repository not on the availability of any underlying system. Finally, with a physical repository, data abstractions can be

0195-4210/95/$5.00

integrated directly with the base data providing the users with unique views of the data[4]. Data abstractions would be more difficult and time consuming to provide dynamically for a logical repository because the process involves complicated procedural code.

The physical repository architecture has drawbacks as well. Unless expensive database replication tools are used, the repository can be a bottleneck and presents a single point of failure.[5] Also, this architecture has an inherent delay in the availability of data and may be more costly because of redundant data storage.[6] With our system, this delay is significant since we only update the repository once a day; but the delay can be reduced significantly with updates to the repository triggered by updates to the underlying systems.

While our repository is not appropriate for decision-support applications that require immediate data, it has proven sufficient for some decision-support applications.[2,3] These applications have been successful at Barnes Hospital, and this success has resulted in requests for these applications at other hospitals.

EXTENSION APPROACHES

We considered three approaches to extending our repository to accommodate data from additional sites. In the first approach, each entity in the global schema of the repository would be extended with an additional property identifying the hospital from which the data came. All data would be in one repository. Queries over multiple sites, such as "How many orders for Ceftriaxone, 1 gram have been given in the last month?", could be posed easily. However, existing applications and data upload procedures would need to be changed to incorporate the hospital identifier property. This approach also assumes that data from other sites can be mapped to the existing schema and that the code dictionaries use the same format.

The second approach is similar to the first in that data from other sites must be mapped to the repository schema; but instead of storing all data in the same database, each site would have a separate database. Each database would have the same base schema without the additional hospital identifier property for each entity. The database name would serve as the hospital identifier. Each database also would have its own code dictionaries, which could have different formats. Queries over multiple sites would be more difficult, but could be expressed with unions. Existing applications would not have to be changed, and they could be ported to other sites with minor changes.

The main drawback to the first two approaches is that the local schema at each site must be able to map into the existing global schema. The third approach is based on a standard query model.[6] In this approach, each site would have a separate database as in the second approach, however, the schemas could be entirely different. A standard query model would sit above these repositories (Figure 1).[1,6] Queries on a standard reference schema would be mapped to the local repositories. Since the reference schema would be defined using a semantic data model, which is more expressive than the relational model, the mappings between the local schema and the reference schema would be simpler than mapping between a local schema and an existing relational repository schema. As in the first approach, existing applications would need to be changed to use the reference schema, but upload applications would not need to change. An important advantage of this approach is that the portability of the decision-support applications is not dependent on the local sites using a common repository schema. However, for this approach to be most useful, there must be a standard reference schema of clinical data. There has been work to define a common data model, but this work is in the preliminary stages.[7]

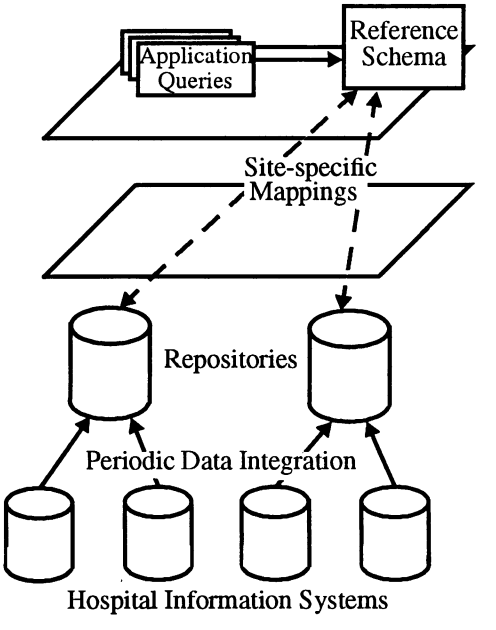

Figure 1: Global Query Model Approach.

We chose the second approach for adding data from Jewish Hospital to our repository. Based on initial studies, we felt Jewish Hospital data could be mapped to our repository schema. Also, we needed to minimize changes to existing code because of our limited resources. As we add other hospitals to our repository

though, it may be imperative to use the third approach if our repository schema proves insufficient.

MAPPING ISSUES

Since we were creating a new database for Jewish Hospital data, we could change the schema; but we wanted to make as few changes as possible, especially to the base schema, to minimize changes to the decision-support applications when they are ported to this database. In the end, we were able to map the local schema to our repository schema without any changes to the base schema, but several challenges presented themselves along the way. The issues we addressed included:

- syntactic and semantic differences in the data, such as different representations of data, different data types for properties, different dictionary formats, and different processing semantics;
- missing and additional data;
- the use of various standards for clinical data; and
- where to do the mapping.

Syntax and Semantics

One of the first issues we addressed was the difference in patient identifier representations between the hospitals. At Barnes Hospital, each patient is assigned a unique 14 digit identifier based on his social security number. In addition, a nine digit unique registration number is assigned to a patient for each admission to the hospital. Demographic data such as name, age, and sex are associated with the patient identifier; and data about a patient's stay in the hospital such as lab results, medications, and rooms occupied are associated with the registration number. At Jewish Hospital, each patient is assigned a unique identifier, but it has only seven digits and is not associated with the patient's social security number. Instead of assigning a new identifier for each admission, however, Jewish Hospital associates a two digit sequential encounter number and a two digit insurance identifier with the patient identifier. The mapping of the patient identifier to our global schema is straightforward: assign the seven digit identifier to a 14 digit field assuming leading zeros. To map to the registration number, we combined the patient identifier with the encounter to create a nine digit registration number.

The problems with this mapping include losing the insurance information and removing the explicit nature of the encounter. Also, we do not have a link between a patient's identifier at one hospital and his identifier at the other hospital. Without this link, users cannot retrieve cross-hospital data on a patient without either knowing both identifiers or using demographic data for the link, which is much less reliable. To solve this problem, we may eventually incorporate the Master Patient Index (MPI) being developed at BJC Health System. The MPI will comprise a unique identifier for each patient seen at any member institution, links to the identifiers for this patient at any other member institution, and basic demographic data for matching and searching.

Similar to the identifier representational difference, we encountered data type differences in several properties in the schemas. For example, in the global schema a patient's race is coded as an integer. A dictionary table relates the code to a character abbreviation and race name. At Jewish Hospital, race is coded as a one or two character abbreviation for the race, and the dictionary associates the abbreviation to the race name. For the race property and other similar cases, we assigned integer codes to the Jewish Hospital abbreviations and mapped the incoming abbreviations to these integers. For other cases where no dictionary was involved, we converted numeric values into character strings, such as the dosage amount for a medication. Fortunately, other than the dictionary cases, we did not encounter any cases where the global schema represented a property as an integer and Jewish Hospital represented it as a character string. Otherwise, we would have been forced to create a new dictionary or modify the base schema.

As with the race codes, some of the dictionaries for coded information had different formats at Jewish Hospital. For the race dictionary, we were able to map it directly to the global schema dictionary by creating integer codes for each race. For other dictionaries such simple changes were impossible. We modified the dictionary schema to handle the differences.

The medication formulary is one example of such a case. The formulary comprises the codes, classifications, and names for the various medications used at the hospital. The code is then used in the medication orders to identify the appropriate medication. Both hospitals use this approach, but they differ in the amount of information that a code identifies. Codes in the Barnes formulary identify a medication and the route of the administration, such as oral or intravenous; whereas codes in the Jewish formulary additionally identify the normal dosage of the medication. At Barnes, each medication order includes the dosage to be administered, thus dosage is a property of the order not the formulary in the global schema. A medi-

cation order at Jewish assumes the normal dosage in the formulary unless an overriding dosage or dosage multiplier is supplied in the order. We modified the formulary in the Jewish database by adding the dosage data; but we maintained the semantics of the medication orders in the global schema by adding the dosage data from the formulary to the orders as they were loaded. Modifying a dictionary by adding properties causes fewer problems than other modifications, because additions do not affect existing applications and users only need to learn the additions instead of two semantically different dictionaries.

Differences in dosage were not the only semantic differences we encountered with the Jewish medication orders. Processing of orders is quite different at the two hospitals. At Barnes, each order is assigned a sequential order number unique to a patient. Any modifications to this order, such as discontinuing it, are made directly to the order. Jewish Hospital also assigns unique sequential order numbers for a patient's orders, but any modifications are recorded as new orders. For example, when an order is discontinued, a new order is created as a discontinue order. The original order is marked discontinued, but only the new order contains the discontinue date and time and who discontinued it. This same process occurs with any renew and hold orders.

To handle these processing differences without modifying the semantics of the global schema medication orders, we store only the original orders and update these orders with data from the matching update orders as they occur. There are two problems with this approach. One, we have not been able to find a direct link between original orders and update orders. Without this link we use a best match algorithm in which we may mismatch orders and make inappropriate updates. This problem has been rare but uncomfortably present. Two, we lose the semantics of the local schema processing such as the update order numbers.

Missing and Additional Data

Besides syntax and semantics, we addressed the issue of missing and additional data. The global schema was based on the data electronically available from one institution. Electronic availability of data can vary greatly between institutions, though. In our case, there was an overlap in the data available from the two hospitals. We were able to obtain most of the data necessary for the global schema from Jewish Hospital. The items missing were ICD9 diagnosis and admitting physician. Fortunately, missing these data was of little consequence, since they were optional in the schema. Additional data should be of no concern, since it can be ignored. However, additional data can also enhance the usefulness of the repository. The additional data available from Jewish Hospital was patient temperatures. In one of our decision-support applications, temperature played a decisive role but had to be gathered by the user from the patient's paper chart. We incorporated the temperatures into the global schema without modification to the schema by treating temperature as another patient measurement, such as height and weight. The only change necessary was adding measurement codes for the various types of temperature (e.g., oral, rectal, etc.).

Standards

While the previous issues hampered mapping, the use of clinical data standards by the two hospitals aided mapping. With standard codes, mapping, developing, and querying are simplified. Only one dictionary needs to be defined, used, and maintained. For example, both hospitals use the AHFS medication classification in their formularies, and both have a link to the NDC medication codes, although Jewish Hospital augments these codes. When a user queries the repository for data on a certain classification of medications, such as antibiotics, she only needs to know the standard classification scheme not a unique scheme for each database. While they did employ some standards, many more are needed. Standards are needed for patient identifiers[8], lab tests, physician codes (such as Medicare codes), service codes, medication frequency and route codes, and others. Without these standards, each database will have different dictionaries, increasing the learning curve for users and the maintenance costs for administrators.

Physical Mapping

Once the logical mappings from the local schema to the repository schema were determined, we needed to decide where to perform these mappings. Figure 2 illustrates the migration of the data from Jewish Hospital information systems to the repository. Each arrow in the figure, except for the file transfer arrows, represents a potential site for mapping the data.

While some mapping is performed at each of these steps, the majority of mapping occurs in the final step. In this model, most changes to the mapping are localized to a few SQL scripts, and the original data from the local schema are kept mostly intact in the files, which are archived, which serves two purposes. If a new mapping, new schema element, or new approach is created, the archived files of original data can be used to repopulate the database.

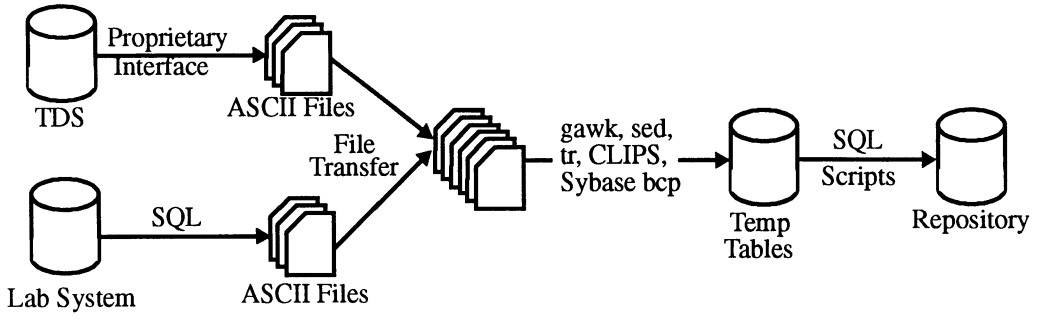

Figure 2: Data Migration Steps.

DISCUSSION

A significant effort was required to add Jewish Hospital data to our existing clinical data repository. We believe, however, that the approach we took minimized this effort. By creating a new database with the same base repository schema, the existing repository was unaffected, and thereby existing upload procedures and applications were unaffected. Also, porting the decision-support applications to the new database will be quite simple since no changes were made to the base schema, syntactically or semantically.

While this approach minimized the work, it could have been simpler if more forethought would have been used when designing the initial repository. Our focus was too narrow when we created the repository for Barnes Hospital data. Although we minimized our work for this extension, we may be making the same mistake. The local schemas of other hospitals that potentially could be added to the repository may not map to our global schema. If this problem occurs, we will be forced to modify the global schema or use the approach proposed by Sujansky[6]. In either case, significant changes will be required of our applications.

As was the case in creating our initial repository, the most difficult aspect of adding Jewish Hospital data was discerning the semantics of the data in the underlying information systems. Poor documentation and proprietary systems hindered this task considerably. However, having a predefined global schema and previous experience in this task simplified the work.

ACKNOWLEDGMENTS

Gary Meyer wrote the extract and parsing programs for retrieving data from Jewish Hospital. This work is supported by NLM Grants 5-R29-LM05387 and U01-LM05845 and by funds provided by the Department of Pharmacy, BJC Health System.

References

1. Marrs KA, Steib SA, Abrams CA, Kahn MG. Unifying Heterogeneous Distributed Clinical Data in a Relational Database. In: Clayton, C, ed. *Proceedings of the Symposium on Computer Applications in Medical Care*. New York, NY: McGraw-Hill, 1994:644-48.
2. Kahn MG, Steib SA, Fraser VJ, Dunagan WC. An Expert System for Culture-Based Infection Control Surveillance. In: Clayton, C, ed. *Proceedings of the Symposium on Computer Applications in Medical Care*. New York, NY: McGraw-Hill, 1994:171-75.
3. Abrams CA, Steib SA, Marrs KA, Jepsen K, Kahn MG. An Expert System for Appropriately Dosing Renally Excreted Drugs. Submitted to *Symposium on Computer Applications in Medical Care*, 1995.
4. Kahn MG, Marrs KA. Creating Temporal Abstractions in Three Clinical Information Systems. Submitted to *Symposium on Computer Applications in Medical Care*, 1995.
5. Deutsch L, Fisk M, Olson D, Bronzino J. Building a Children's Health Network: City-wide computer linkages among heterogeneous sits for pediatric primary care. JAMIA Symposium Supplement 1994:536-40.
6. Sujansky W, Altman R. Towards a Standard Query Model for Sharing Decision-Support Applications. JAMIA Symposium Supplement 1994:325-31.
7. ANSI/HISPP MSDS JWG for a Common Data Model IEEE P1157 Medical Data Interchange Working Group. Trial-Use Standard for Healthcare Data Interchange--Information Model Methods. June, 1994.
8. Hieb BR. A Proposal for a National Health Care Identifier. JAMIA Symposium Supplement 1994:469-72.